N1979

Document No. S-12

MAP™

GUIDE TO THE MAP FAMILY OF I/O INTERFACES

cspi CSP Inc.
The leader in Digital Signal Processing

MAP the only Array Processor that functions in real time—all the time

The MAP Series of floating-point array processors and programmable I/O interface Scrolls combine powerful arithmetic capabilities with outstanding I/O performance. Greatly increased processing power is now available using a unique multiprocessor architecture that allows non-interfering data input and output at rates up to 36 Megabytes/second. Direct access to MAP memory eliminates data traffic jams to and from the host computer, leaving the host free to do higher level tasks.

Array Processing with MAP

MAP Three Bus Architecture—Essential to the MAP throughput performance is the internal triple bus structure. Physically separate memory banks containing up to 256K bytes each are available to all arithmetic processors and I/O Scrolls.

Arithmetic Capabilities—Properly rounded and normalized 32-bit floating-point arithmetic is carried out in parallel multipliers and adders, with data fetches from MAP memory overlapping the arithmetic.

Low-Overhead Data Transfers—The host delegates transparent, non-interfering, data transfers to and from external devices to eliminate most I/O overheads. The direct access memory as opposed to interleaved memory bus architecture achieves full speed regardless of program design or storage allocation.

Fast Programmable Data I/O—Direct MAP memory accesses by external devices are handled by one or more programmable I/O Scrolls at data rates up to 36 Megabytes/second.

Extensive Software Support All processors are supported by high level SNAP-II FORTRAN calls. SNAP-II routines are available for commonly used array processing and I/O functions including FFT's, digital filtering, matrix manipulation and continuous or block acquisition of data. FORTRAN definable function lists allow the host to delegate conditional execution of a sequence of commands in the MAP without host supervision. No need for an additional level of time-consuming function chaining, assembly, and linking. The comprehensive "IF," "FOR," "WHILE," and "CASE OF" conditional execution structured commands are available from the host FORTRAN system.

Direct Map I/O for Array Processing Power

With its internal executive program, multiple bus architecture and programmable I/O interfaces, MAP is smart enough to get its own data—an important feature that avoids contention for host resources and the accompanying overhead burdens which arise when data is passed via the host computer.

Why Delegate Routine I/O to MAP?

Delegating routine I/O to MAP with an I/O Scroll permits the full arithmetic power of MAP to be utilized. The arithmetic throughput of array processors has long exceeded 100,000 samples per second even for complex calculations such as the Fast Fourier Transform. However, the typical host computer is severely loaded in attempting to maintain 100 KHz continuous data flow, and in most instances it is not even possible. The typical processor must idle while the host computer reads out processed data and writes in raw data.

The MAP I/O approach, which supports high-speed I/O data transfers from external sources directly into the array processor memory, provides for more effective use of the arithmetic resources. The MAP I/O Scroll and peripheral interfaces, which are designed as a family, allow transparent data transfers to be carried out for a wide range of devices at data rates up to 36 Megabytes/second.

How Do I/O Scrolls Work?

An I/O Scroll is a programmable, bi-directional interface between an external device and a MAP memory. Each Scroll can be easily programmed for calculating addresses, counting, resetting itself, and notifying other processors of its progress. Since a Scroll can restart itself and then signal the other processors that a buffer is full, no samples are lost during high-speed double buffering. The alternative hard-wired approach typically requires a host computer interrupt to reset the I/O Controller between blocks of data, and inevitably results in lost data samples. Not so with MAP. FORTRAN IV calls can be used to execute standard I/O programs using the SNAP-II library, thereby reducing software development time and costs.

I/O Scroll Configuration Flexibility

Multiple Scrolls of different types may be mixed within a single system. This configuration flexibility allows the user to change his overall system characteristics to suit changing requirements simply by adding boards in the field. To allow custom hardware interfaces to be configured, one member of the Scroll family—the IOS-2—is available with board space allocated for user designed logic. Program flexibility is emphasized in the IOS-3 which is built around a microprocessor having an extensive and powerful instruction set. The IOS-4 is a high-speed interface which can feed two memory banks in parallel with 32-bit words at the transfer rate of 36 Megabytes/second, or a single memory bank at a transfer rate of 18 Megabytes/second.

Are There Standard Peripherals Already Interfaced Through a Scroll?

Yes! For example, the Analog Data Acquisition Module (ADAM). ADAM simultaneously samples up to 16 channels of analog data and converts the samples to digital form at aggregate rates up to 270 KHz with 12-bit resolution. It permits multi-rate sampling, external sampling, sophisticated triggering, and programmable data multiplexing patterns, all from simple SNAP-II FORTRAN calls.

Other specialized Scroll supported I/O devices include the AOM (Analog Output Module) for generation of analog signals, and the custom Megadisk interfaces for direct MAP access to popular disk storage units.

Doesn't That Take A Lot Of Host Computer Supervision?

Not at all. Each MAP contains a Central System Processing Unit (CSPU) and the SNAP-II Executive. This hardware/software combination responds to SNAP-II FORTRAN-IV calls, and supervises the Scrolls and MAP processors. The host need only start the action. Powerful software features, such as the MAP "WHILE" command, allow the host to go away and do other things.

Software Support

The MAP family of I/O interfaces is fully supported with comprehensive development and debugging software described in this section.

FORTRAN Support Routines

Standard library routines include a wide range of general purpose array processing functions. These are found in the SNAP-II FORTRAN library which contains a variety of mathematical, control and I/O routines, all callable as functions or subroutines from a host FORTRAN program. Enhanced FORTRAN features are available such as function list definition and conditional execution of lists according to "FOR", "IF", "WHILE", and "CASE OF" commands. Once these function chains have been defined and initiated in the user's FORTRAN program, execution proceeds within the MAP alone and the host is released to do higher-level tasks.

When programming is carried out using standard SNAP-II library functions as shown below, the full capabilities of the host operating system, including compiler and editor, can be exploited. No second layer of array processor code chaining, assembly and linking is necessary, a factor that is particularly important in the development phase when application programs are subjected to frequent changes.

Modular Software Development

At the assembly language level, distinct modular programs are developed for the arithmetic unit, addresser unit, CSPU and I/O Scrolls. This separation of tasks greatly simplifies development of any program which must be written in assembly code. With this basic approach programs can be streamlined to suit a particular application problem, thereby reducing all overhead functions to a bare minimum, or they can be written in general form to suit multiple users.

Custom assembly language routines can be embedded in standard Executive support routines to make them available as higher level language (e.g. FORTRAN) library calls if desired. Well documented procedures for achieving this integration are available. This philosphy gives the user access to the full power of the arithmetic and I/O processors in the MAP as well as the luxury of a higher level language to handle the more tedious programming tasks such as operator parameter input, performance monitoring and resource allocation. No compromises in performance are necessary in this programming mode because individual assembly language routines can be made as tight and specialized as is desired.

MAP System Software

The MAP support software supplied by CSPI includes the MAP Executive program and a complete set of library SNAP-II routines. The utility routines, provided for program development and debugging, consist of a cross assembler, simulator, loader, and debug package. These programs are written in standard ANSI FORTRAN-IV for the host.

CSPI Application Analysis and Programming Services

Full application and program development support services can be provided by arrangement. CSPI maintains a staff of technical experts who are available as consultants to undertake the design of specialized software and hardware modules for specific customer needs.

SNAP-II FORTRAN EXAMPLE

This example illustrates the way in which data acquistion and display functions could be programmed with MAP. Data samples are acquired using SNAP-II calls, and output to a Scroll after a discrete filtering array operation. Single buffering is used for simplicity of presentation.

```
C.....SUBROUTINE TO START MAP PROCESSING LOOP

      CALL MPDRB(CODE,1,36.)          • Define buffer, Bus 1, for Scroll code
       :
      CALL MPDRB(OUT,3,1024.)         • Define real buffer for output on Bus 3
      CALL MPDRB(IN,2,1024.)          • Define real buffer for input on Bus 2
      CALL MPWDB(CODE,HSRT,0,1,HEND)  • Transfer Scroll driver from host to MAP
      CALL MPLDS(SCROLL,2,CODE)       • Load Scroll program into Scroll.

C.....DEFINE FUNCTION LIST TO BE EXECUTED LATER

      CALL MPBFL(LOOP)                • Begin function list definition.
           CALL ADM1D(RATE,MODE,CH1,IN) • ADAM samples data input to buffer IN
           CALL DFL22(OUT,SA,IN,SB)   • Discrete filter data, 2 poles, 2 zeros.
           CALL MPRNS(SCROLL,2,OUTPUT) • Run Scroll to output data.
      CALL MPEFL                      • End function list definition.

C.....EXECUTE FUNCTION LIST IN MAP

      CALL MPIWL(I1,0,I2,LOOP)        • Loop on function list in MAP until I1 = I2.
       :
      RETURN

C.....HOST  NOW CONTINUES OTHER TASKS
```

The "WHILE" function list execution continues indefinitely until one of the integer values is altered. Details of the array function calls may be found in the SNAP-II Programmers Reference Manual and Rev. 3 Release Notes.

Printed in U.S.A.

Data flow in a Computer/Array Processor System

The way in which data flows into and out of an array processor system can significantly affect its overall efficiency and real-time capability. As a comparative example of both the traditional approach and the MAP approach to I/O described below, consider an application requiring four analog signals to be digitized, Fourier transformed and displayed with no-samples-lost. At a hypothetical channel sampling rate of 25 KHz, the total input rate is 100 KHz. In this example, data are segmented into blocks of 1024 samples, transformed with an FFT routine, and output to an x-y display at the same 100 KHz rate.

THE TRADITIONAL APPROACH
Routine I/O Handled By Host Computer

With the configuration shown below, data acquisition and display peripherals are interfaced to the host computer and data must be transferred to the array processor via the host memory. Each sample utilizes four cycles of host memory as follows: ☐ A/D to the host memory ☐ host memory to the array processor ☐ array processor to the host memory ☐ host memory to the display.

Thus, the host memory is being utilized at a 400-KHz rate for very routine I/O. Considering demands for memory cycles alone, it is apparent that most of the available host memory cycles are consumed by these transfers. In fact, when software overheads for setup of the three direct memory access controllers are accounted for, the host I/O capabilities will very likely be saturated. In the case of simpler single-bus "vector box" types of array processors, this situation is further aggravated by I/O cycle stealing demands on the array processor memory. As a result of these effects, the goal of continuous throughput at 100 KHz will usually be unattainable.

Disadvantages ☐ With the traditional approach, the host computer becomes saturated as a result of I/O transfers at quite modest data rates. ☐ Single bus array processors are performance limited by memory contention bottlenecks when concurrent arithmetic and I/O operations are attempted.

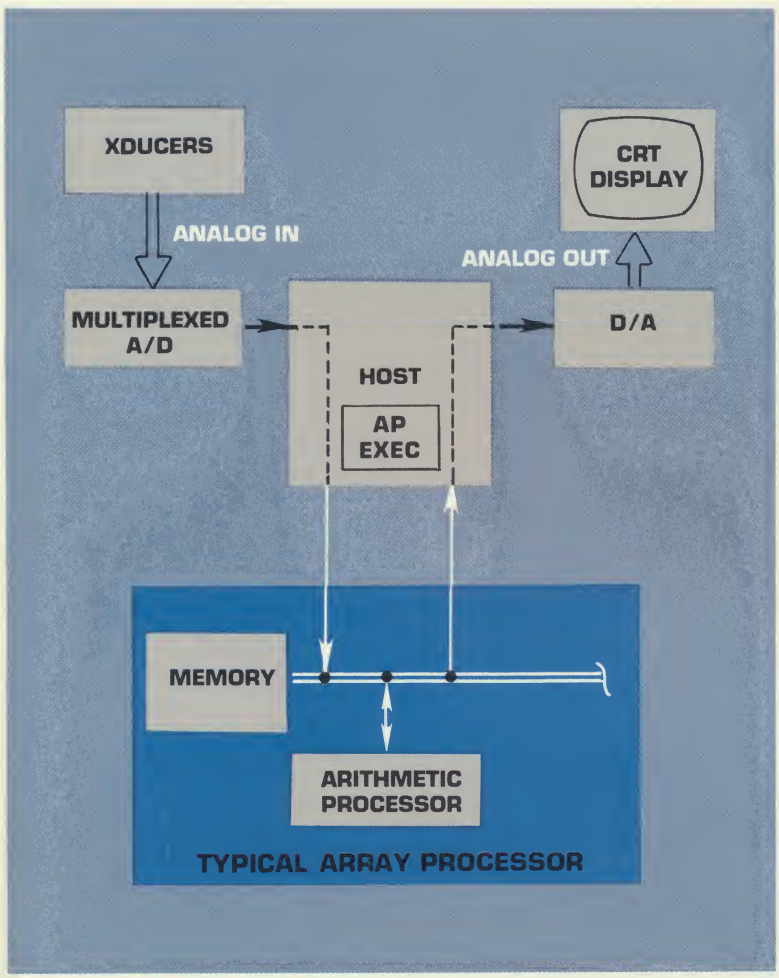
THE MODERN APPROACH
Routine I/O Delegated To The MAP

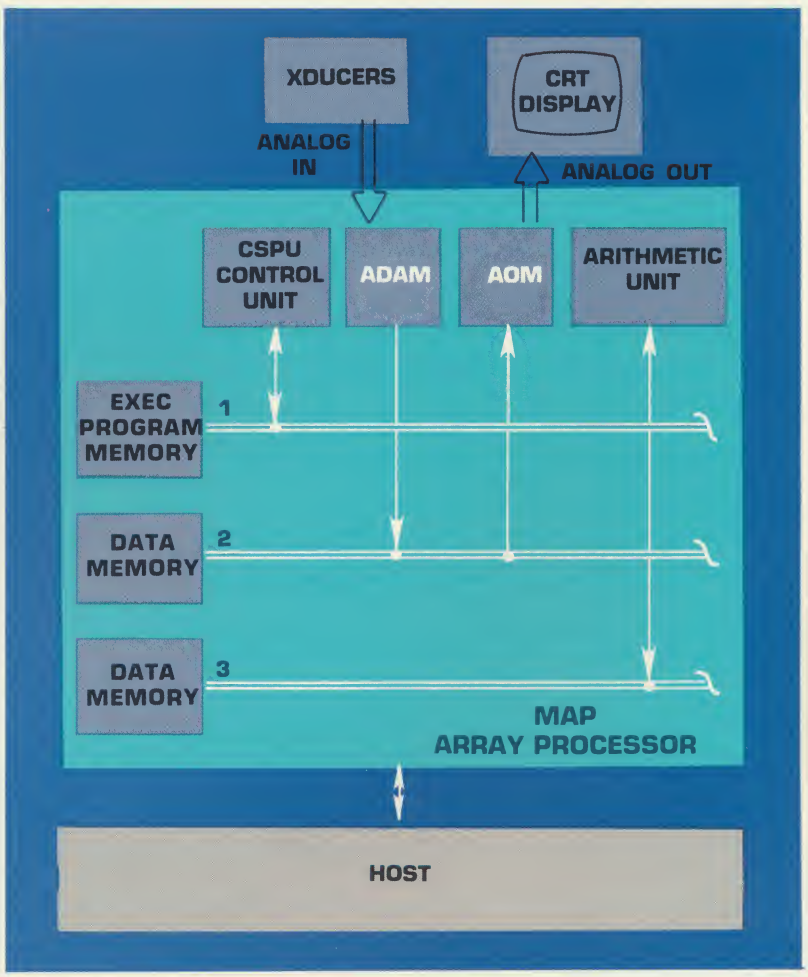
Using the scheme below, the host can delegate complex data acquisition, signal processing, and display operations to the MAP. Since the array processing programs reside in the MAP, the host computer is released for interaction with the user to modify parameters or perform other tasks. Data flows through the system on independent buses in the following way: ☐ samples from the A/D may be transferred to MAP memory Bus 2, ☐ arithmetic operations are carried out on earlier data on Bus 3, ☐ previous results are output to the display from a temporary buffer.

The memory bus utilization is switched when all of the above parallel operations have been completed. I/O then takes place on Bus 3 and processing continues on Bus 2.

For a typical FFT time of several milliseconds, a continuous throughput sample rate in excess of 200 KHz can readily be achieved using one ADAM and one AOM to handle I/O operations.

Advantages ☐ The host can delegate routine I/O tasks to the MAP, and provide additional supervisory or processing services to the user. ☐ Direct MAP memory access removes the host-array processor I/O bottleneck. ☐ Separate MAP memory buses ensure completely overlapped non-interfering I/O and arithmetic operations.





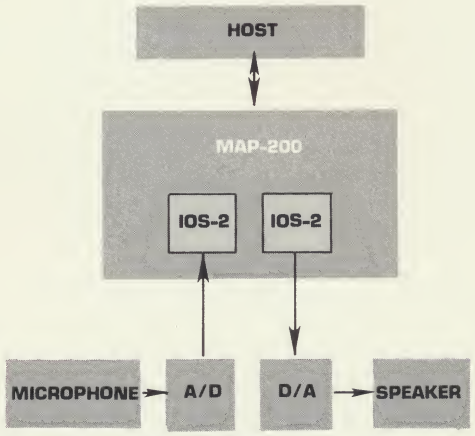
How CSPI Customers are using I/O Scrolls

CSPI customers have been using I/O Scrolls to interface a wide variety of devices to their MAPS. The three customer-system summaries below were chosen from among many. Notice that the range of requirements vary from a one sample output for each sample input —in the case of the speech processing application, to the input of 96,000 samples resulting in the output of two tracking coordinates—in the case of the TV tracking application.

Speech Processing at an East Coast University

In this system, the MAP-200 performs real-time speech enhancement at a continuous sample rate of 18 KHz with no samples lost. The processing depends on an adaptive filtering technique to remove interferring signals and noise from the desired signal. Once the process has been initiated by the host computer, no further supervision is required, and even the filter coefficients are selected under internal MAP control. The digitized input samples and quantized filter parameters enter the MAP directly through an IOS-2 Scroll, and the resultant output goes to a speaker and display unit via a second IOS-2. No host computer memory space or access time is required.

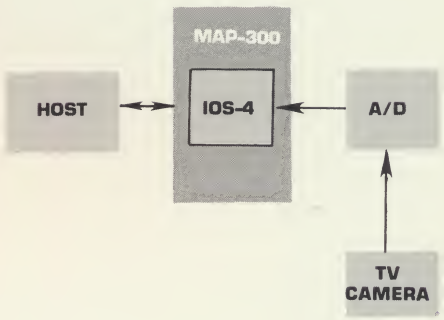
- ☐ **Continuous Enhancement**
- ☐ **18-KHz Sample Rate**
- ☐ **Parameters Controlled by CSPU**
- ☐ **No Data to Host**



Video Tracking, West Coast Aerospace Contractor

Eight-bit samples enter the MAP directly from the video digitizer through the I/O Scroll (IOS-4) without external buffering. The IOS-4 sets up the necessary double buffering for lines of the sampled video data, and directs the arithmetic processor to the appropriate data locations. An object in the field of view is tracked by means of a centroid calculation performed on each line of data, and computed results are returned to the host 30 times a second following each video frame. A single 480-element line is processed in real time during the scan and fly-back time for the succeeding line of the 200-line picture. No data samples are missed, and optimal MAP memory efficiency is maintained by packing four samples to a 32-bit MAP word. Host supervision is not required, and no host memory time or space is necessary for data handling.

- ☐ **Ultra high-speed Tracking**
- ☐ **8 MHz Sample Rate**
- ☐ **Direct Byte Addressability**

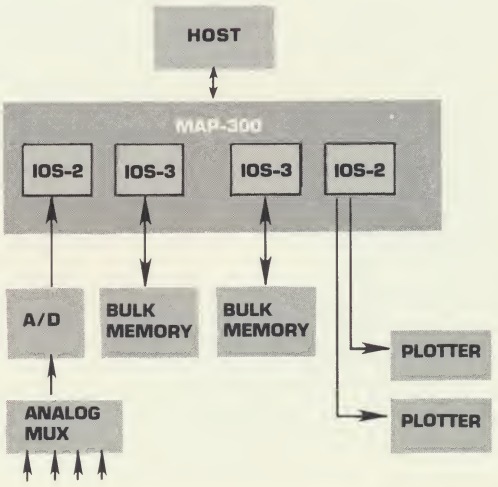


Sonar Signal Processing, Defense Contractor

In this application, sonar signals are multiplexed, sampled and written into MAP internal memory by an IOS-2 Scroll. An IOS-3 then transfers it to an external bulk memory and later reads the data out in a different order in preparation for spectral analysis. The MAP-300 performs FFT's, and averages several records using the second bulk memory for temporary storage.

Finally, data records are accessed by an IOS-2 for plotting purposes. In this manner, data input, processing, and output are accomplished independently of the host computer. The user, through the host, directs record selection for plotting and determines parameters such as averaging time.

- ☐ **256 Channels Sampled at 400 Hz**
- ☐ **Two External Bulk Memories**
- ☐ **Two Plotters For Output**



Place Postage Here

CSP Inc.
209 Middlesex Turnpike, Burlington, Massachusetts 01803
U.S.A.

Please fold and staple before mailing

Use this convenient Reply Card for Additional Information Quotation or to arrange a Design Meeting.

ACTION REQUEST

Yes, I'm interested in unburdening my host computer by delegating routine I/O to MAP:

- Send me more information on FORTRAN controlled . . .
  - ☐ I/O Scrolls for interfacing digital devices. I want to interface to _____
  - ☐ Analog input and output modules. I have the following signals _____
  - ☐ Megabyte mass storage interfaced to MAP. My storage requirements are _____
- Critical considerations are:
  - ☐ Minimizing host supervision.
  - ☐ No-samples-lost continuous processing.
  - ☐ High-peak transfer rates.
  - ☐ Performing I/O and array processing from FORTRAN.
- My host computer is _____, operating system _____
- My requirement is ( ) immediate, ( ) _____months, ( ) future.
- Other information required _____

Name _____

Title _____

Company _____

Address _____

Phone _____
Area Code Number

TELEX _____

M904

MAP™ 6400

the World's first 64-bit Array Processor for Scientific and Engineering Applications . . .

cspi CSP Inc.

The leader in Digital Signal Processing

~1979

The MAP-6400

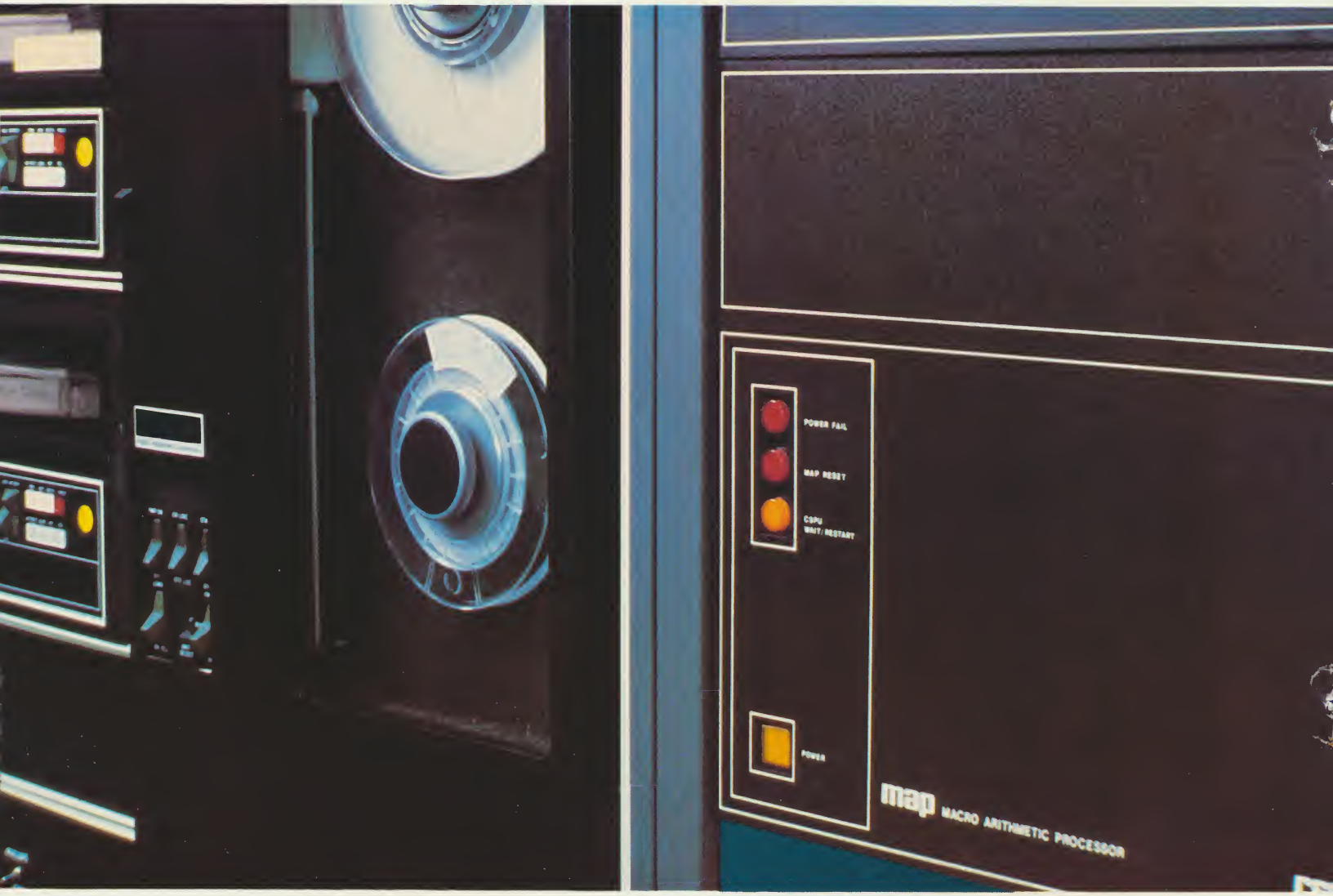
After more than a decade of evolution, array processor circuit technology, architecture and software concepts have matured into a product of compelling interest to those involved with scientific and engineering problems. The MAP-6400, designed as a minicomputer peripheral for high-speed arithmetic performance, leads the way with 64-bit (double precision) floating-point arithmetic capabilities equivalent to those of a multimillion-dollar mainframe processor.

Interactive Laboratory Computing

The economy of the MAP-6400 is in accordance with today's computer cost trends. With its high performance and affordability, MAP-6400 is the obvious answer for users carrying out intensive arithmetic computation on a costly, unapproachable mainframe computer. Now, using a minicomputer equipped with the MAP array processor, highly interactive on-line computing may be realized economically in a lab environment at "mainframe" speed.

Array Processors at Work

In contrast to conventional computers, the array processor uses parallel hardware to overlap the data fetch and store operations with instruction decoding and execution. Sophisticated arithmetic circuits working in parallel achieve many concurrent floating-point multiply and add operations every microsecond. Arithmetic speeds range from 10 to 100 times faster than a mini or supermini, and compare favorably in performance with highly respected multi-processor mainframe computers costing 30 times as much as the MAP/minicomputer combination.



Double Precision for Demanding Applications

All internal data storage and arithmetic operations are in IBM 64-bit format. The 56-bit mantissa maintains 16 full decimal digits of precision through a computation. But it doesn't stop there; multiple high-speed arithmetic units carry out proper unbiased rounding after every operation without sacrificing any of the arithmetic throughput capacity.

High-Speed Precision Computing

Before the MAP-6400, many routines such as matrix inverse and eigenvalue calculations were unstable on a conventional single precision array processor. With double precision host computer arithmetic, such calculations were notoriously slow. Now these double precision calculations can be accomplished at higher speeds than the single precision counterparts in the host.

Precision Margin Makes Programming Easier

The designed-in speed and 64-bit (double precision) format arithmetic of the MAP-6400 provide unequivocal user advantages. Choice of precision is no longer a sensitive and agonizing compromise between speed and accuracy. The 16-digit representation of MAP-6400 data establishes a comfortable precision margin for the user.

Increased Interactive Versatility

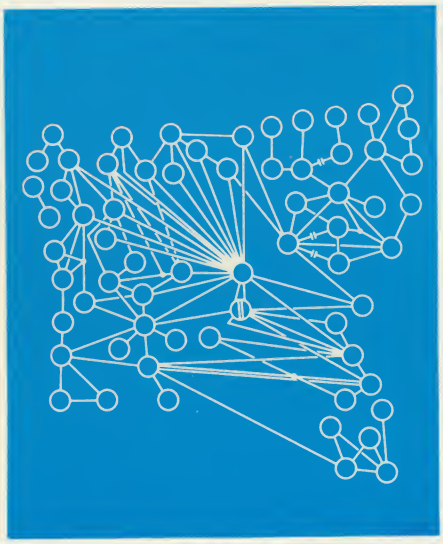
The transfer of a user's application program from a large mainframe computer to a laboratory minicomputer environment brings unexpected benefits in the form of increased programmer interaction. Examples follow.

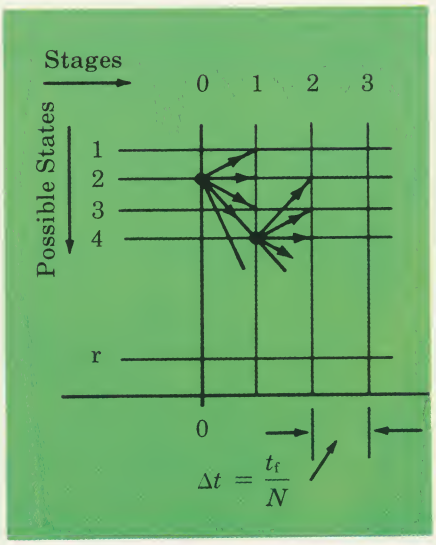
For **non-linear structural analysis,** approximations are often built into the algorithm to ease the computational burden. However, the validity of these approximations must be reassessed at frequent intervals throughout a run to avoid grossly erroneous results. With the availability of MAP-6400, a user is able to periodically evaluate performance and modify his strategy as the task is executed. For this kind of problem, a mainframe would not allow the hands-on accessibility, while the minicomputer alone would be too slow.

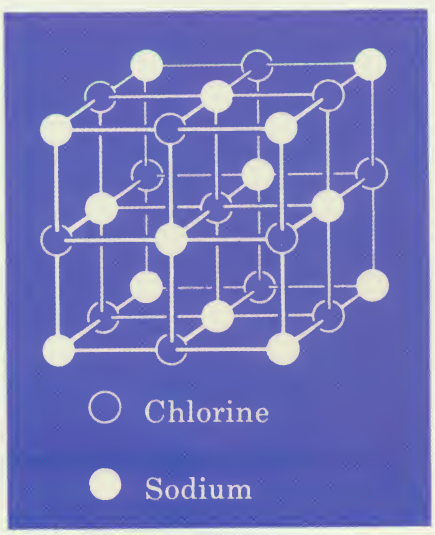
In the world of **molecular dynamics,** the motion of large numbers of atoms and molecules must be described by huge sets of coupled differential equations. The researcher is faced with virtually an infinite number of starting combinations of relative atomic positions and velocities. With real-time interaction a user can be more selective in his simulation choices, and with the MAP-6400 this approach is feasible at a moderate cost.

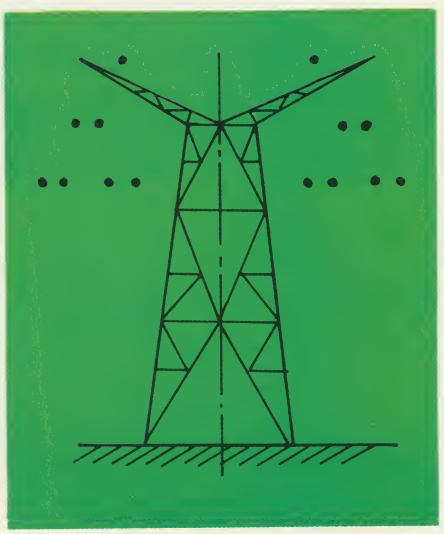
High-Speed Graphics Support

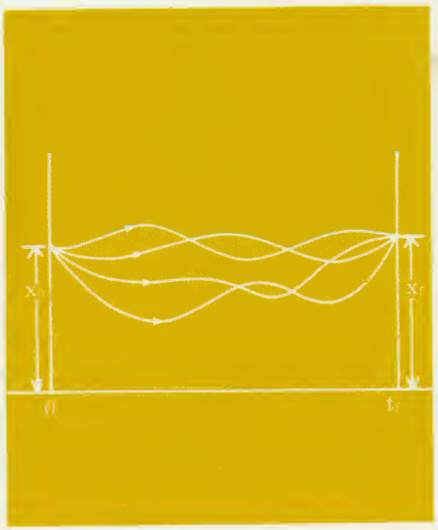
Threaded through almost all of the scientific and engineering applications areas is the requirement for versatile high-speed graphics support. The MAP line excels here too. Three-dimensional image processing features including translation, rotation, zoom, perspective and hidden line resolution can all be implemented in MAP software to modify the data base before it is transferred to a display. With MAP's inherent control and flexible I/O capabilities, direct memory output via a MAP interface board can be achieved.











MAP Application Areas

The MAP-6400 system and its supporting software are aimed at satisfying a wide range of scientific, mathematical, and engineering application requirements, including the following:

Numerical Analysis and Mathematics

- Matrix Operations and Properties
- Linear Equation Solution
- Linear Programming
- Differential Equations
- Partial Differential Equations
- Dynamic Programming
- Integral Equations
- Transcendental functions

Applied Science

- Weather Modeling
- Quantum Mechanics
- Chemical Reaction Kinetics
- Seismic Modeling and Analysis
- High Energy Physics

Engineering

- Network Analysis
- Dynamic Systems Analysis
- Structural Analysis
- Fluid Dynamics
- Optics Design
- Load Flow
- Heat Transfer

Operations Research

- Econometrics
- Supply/Demand Scheduling
- Resource Allocation
- Process Control

Versatile MAP System Configurations

The MAP Family

The Map-6400 processor is a natural result of CSPI's experience and expertise gained in engineering and marketing activities with earlier members of the MAP family. The MAP-6400 is an extension of the MAP-200 and MAP-300, of which there are now over 300 installations worldwide.

Many proven features of this 32-bit precision array processor line have been inherited by the MAP-6400. These include asynchronous processor and memory architectural features which were borrowed from our established high-technology base. And of course, the new member of the family comes complete with its own internal controller so that the host computer can be relieved of routine monitoring and control tasks during the periods of heavy array processor computation. Acting as an intelligent co-processor to the host, the MAP-6400 not only speeds up program execution, but also frees the host CPU for other processing tasks to be carried out in parallel.

Software Compatibility

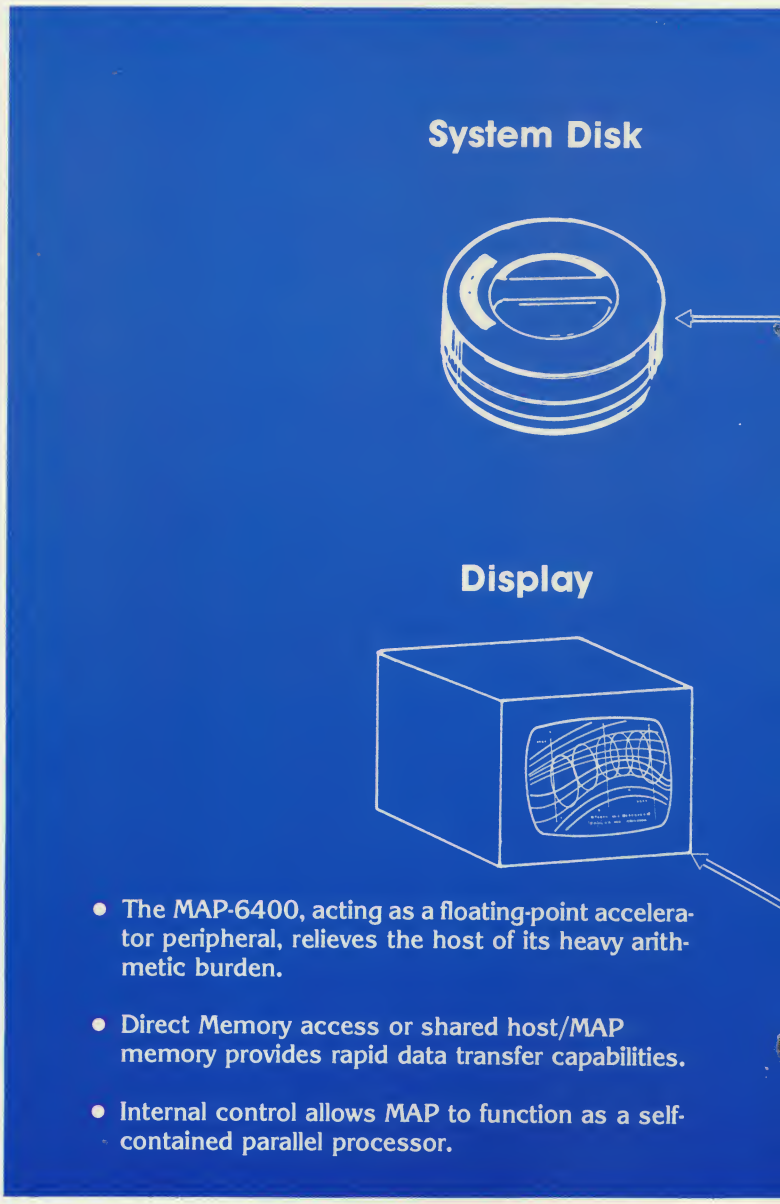
A basic goal in the development of this 64-bit array processor was the retention of software compatibility with the existing MAP line. Full instruction set compatibility has been achieved enabling present 32-bit MAP customers to readily take advantage of the MAP-6400. This benefit represents an umbrella of software support for the user at both the custom assembly code and the FORTRAN subroutine library levels.

Internal MAP Operating System

Within the MAP-6400, all sequencing and control of tasks is carried out by an internal software monitor, the MAP Executive. This sophisticated real time monitor, developed originally for the 200/300 series, has been further refined for use in the 64-bit machine. In conjunction with the host operating system, the Executive maintains the integrity of data areas internal to the MAP, as well as those shared with the host CPU in the host main memory. It is also responsible for multiprocessor task synchronization, including data transfers between the MAP and its host.

Product Reliability

MAP-6400 components are subjected to the same rigorous scrutiny and testing as those used in the 200/300 series. With the implementation of many common design concepts, a high degree of reliability is realized for the MAP-6400 on the basis of past experience with MAPs already in the field. Comprehensive and on-call maintenance agreements as well as central software updating services are offered by CSPI to ensure that an installation is always running at full efficiency.

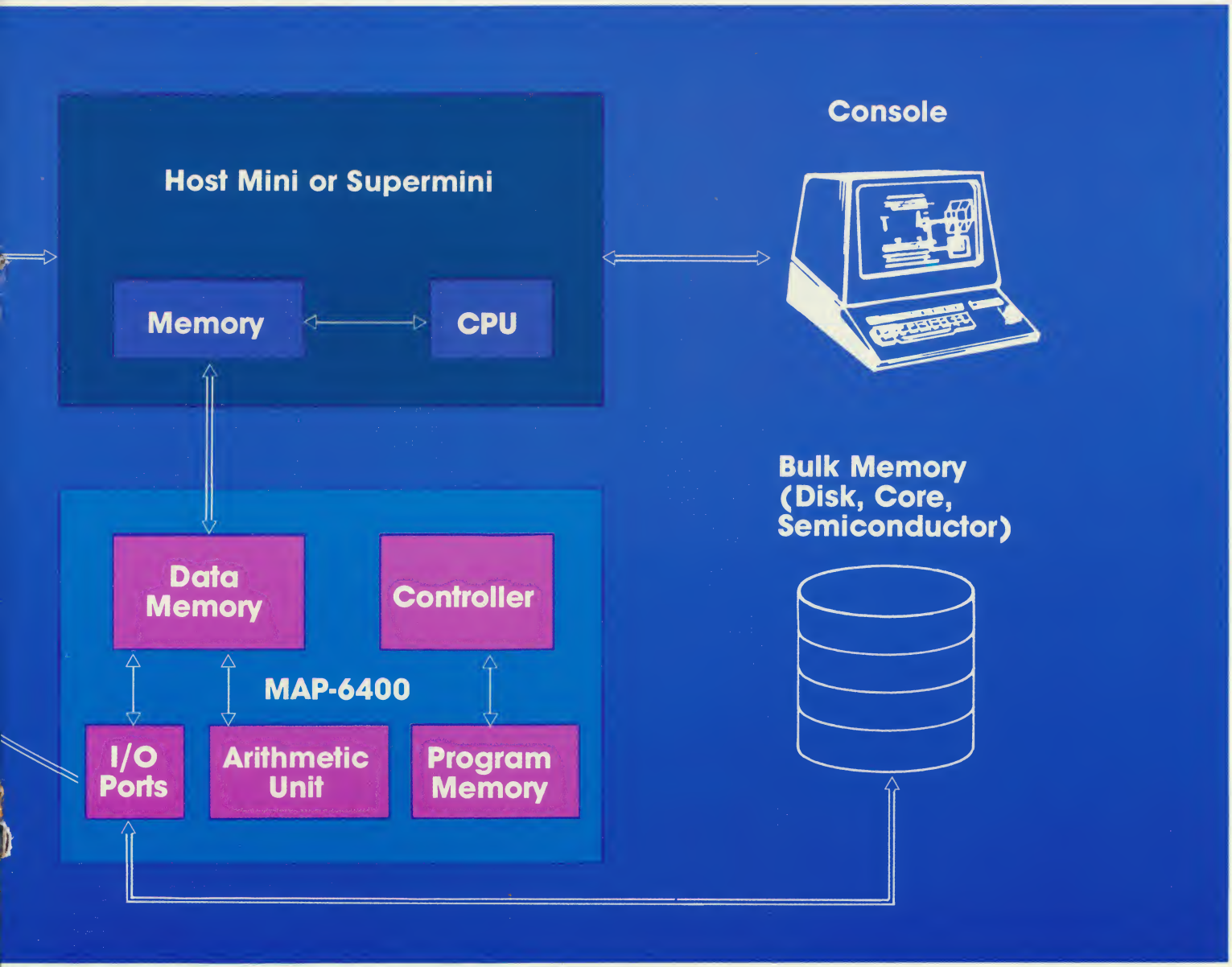


- The MAP-6400, acting as a floating-point accelerator peripheral, relieves the host of its heavy arithmetic burden.
- Direct Memory access or shared host/MAP memory provides rapid data transfer capabilities.
- Internal control allows MAP to function as a self-contained parallel processor.

Hardware Features

The basic MAP-6400 processor unit consists of an arithmetic unit, internal controller, multiple memory system, and interface boards. System options and features are as follows:

- Standard hardware and software interfaces to a wide range of host computers.
- Minimum program memory configurations from 16K words up to 56K words in 8K increments.
- Multi-ported 64-bit data memory in standard sizes of 32K and 64K words (optional parity).
- Data memory expansion beyond 64K using standard bulk memory interfaces.
- Direct access disk storage peripherals.
- Analog input and output peripheral devices to allow direct connection of analog signals.
- A versatile family of programmable digital I/O devices to support custom interface designs.
- Field expandable to accommodate changing application requirements.



Performance Summary

Although detailed performance comparisons are difficult to make, the effectiveness of the MAP-6400 can readily be judged in terms of its computational speed for two familiar arithmetic algorithms.

100 x 100 real matrix matrix multiply 1.0 second
1024 point complex FFT ..... 22 millisec.

When compared with execution times for similar 64-bit calculations on a mini or supermini, the MAP-6400 is seen to be one to two orders of magnitude faster.

For comparisons with the mainframes, cost must be taken into consideration. On the basis of price/performance alone, MAP-6400 ranks with the best and in terms of program execution cost, is an order of magnitude more economical than some. And it is not just the economy—when MAP is installed on your mini or supermini, it can be dedicated to **your** task to achieve throughput rates comparable to or greater than larger machines which must normally operate in a time-sharing mode.

Easy Software Convertibility

To the user, the **real** interface with the MAP-6400 is via the library of MAP host support FORTRAN routines using substitution of subroutine calls at the FORTRAN level. With this method, advantage can be taken of optimized library routines that fully exploit the computational capacity of the MAP-6400. Conceptually then, the software connection looks similar to that for a mainframe scientific subroutine library, or high-speed plotter support package.

The SNAP Library

The library of FORTRAN callable array processing routines, referred to as SNAP, is made available to the user as a standard software product. It currently consists of over 200 routines, and new additions are being incorporated into the library several times a year to satisfy new and expanding application requirements. This library of array functions is fully supported and updated by CSPI.

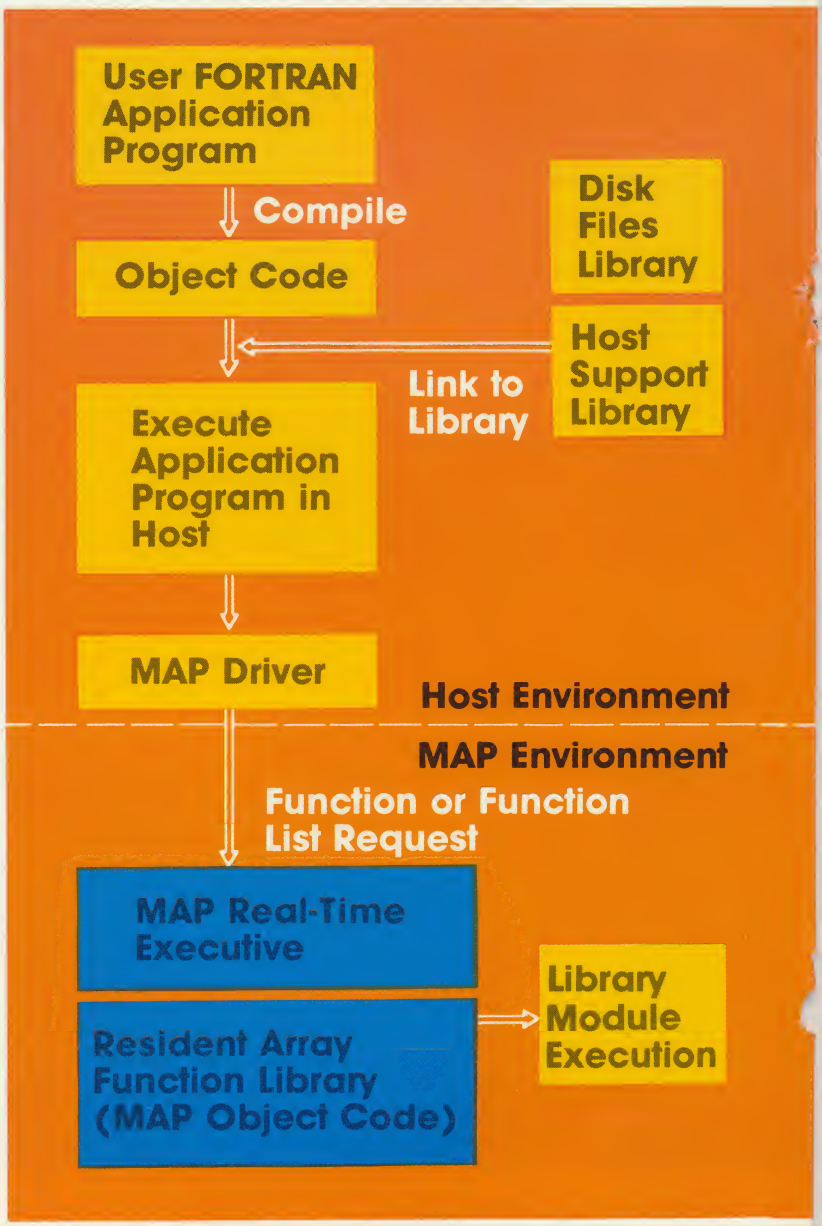
Applications Oriented Software

The current library repertoire includes several hundred routines designed for scientific and engineering problem solving.

- Transcendental function evaluation, and rectilinear/polar coordinate conversion.
- Linear and non-linear vector operations such as weighted vector addition, inner product, square root, magnitude squared and sums of squares.
- Matrix routines with complex and real versions of matrix multiplication, matrix inverse, and generalized matrix properties.
- Efficient one- and two-dimensional Fast Fourier Transforms for real and complex data.
- Solution of linear systems of equations using classical factorization methods; conventional, sparse block diagonal, and general sparse matrix representations.

Utility Software

The software support of MAP is fully rounded out with a variety of packages designed to simplify programming, operation, and maintenance. These standard CSPI utilities include the MAP assembler, simulator and debug routines for program development; the loader for system bootstrap and Executive modifications; and the host loadable diagnostic routines for fault location determination.



The Software Connection

The user communicates with the MAP-6400 via the SNAP host support library. This set of disk resident FORTRAN routines can be thought of in the same way as a general purpose scientific subroutine library. After compilation, when a user job or task is linked, a SNAP library search is carried out by the host operating system. Then, as SNAP subroutine calls are encountered in the application program, the equivalent SNAP host support routines are

MAP-6400 FACT SHEET

64-BIT ARITHMETIC PERFORMANCE/COST COMPARISON

	APPROX. COST ($)	100 x 100 MATRIX-MATRIX MULTIPLY (SEC)	PERFORMANCE/COST (OPERATIONS/SEC/$)
ARRAY PROCESSORS			
MAP-6400	89,000*	1.0	11.0
CRAY-1	7,500,000	0.015	9.0
STAR 100	8,000,000	0.04	3.0
LARGE MAIN FRAME			
CYBER 176	4,400,000	0.42	0.5
IBM 3033	3,070,000	1.1	0.3
IBM 370/168	4,200,000	1.8	0.13
UNIVAC 1100/80	3,500,000	5.0	0.05
SUPERMINI			
VAX-11/780	200,000	12.0	0.4

* Price of host not included.

SPEED COMPARISON WITH MINICOMPUTERS

	FLOATING POINT HARDWARE	30 x 30 MATRIX-MATRIX MULTIPLY (SEC)
MAP-6400	Yes	0.03
SYSTEMS 32/77	Yes	2.03
HP 21MXF	Yes	3.38
HP 21MXE	No	45.0
PDP-11/34	No	47.4

CSP Inc. 209 Middlesex Turnpike, Burlington, MA 01803 · 617-272-6020
The leader in digital signal processing
TWX: 710-332-0835

	INPUT	CORRECT	MAP-6400 ANSWER (LIVE)	HOST DP ANSWER (LIVE)	CRAY-1 ANSWER (STORED)	32-BIT ANSWER
COL 1						
	1/ 1	7	6.99999999	6.99999999	6.99999962	-6.39279
	1/ 2	-168	-167.99999978	-167.99999983	-167.99999331	115.81184
	1/ 3	1260	1259.99999857	1259.99999893	1259.99996084	-642.11792
	1/ 4	-4200	-4199.99999580	-4199.99999688	-4199.99989414	1524.68335
	1/ 5	6930	6929.99999382	6929.99999543	6929.99985419	-1674.25757
	1/ 6	-5544	-5543.99999554	-5543.99999672	-5543.99990045	773.86523
	1/ 7	1716	1715.99999874	1715.99999908	1715.99997324	-90.59256
COL 2						
	1/ 1	1176	1175.99999936	1175.99999961	1175.99996874	425.75220
	1/ 3	-18816	-18815.99998688	-18815.99999195	-18815.99941798	-3399.84985
	1/ 4	105840	105839.99991453	105839.99994751	105839.99644762	4679.02734
	1/ 5	-282240	-282239.99974824	-282239.99984549	-282239.99004366	17580.64062
	1/ 6	388080	388079.99962790	388079.99977199	388079.98585371	-57331.86719
	1/ 7	-266112	-266111.99973043	-266111.99983515	-266111.99007587	57962.43359
	1/ 8	72072	72071.99992376	72071.99995348	72071.99726743	-19916.14844
COL 3						
	1/ 1	-17640	-17639.99998766	-17639.99999217	-17639.99943033	-3181.59888
	1/ 4	317520	317519.99974900	317519.99983999	317519.98945566	19037.88672
	1/ 5	-1905120	-1905119.99836674	-1905119.99895815	-1905119.93594340	59971.56250
	1/ 6	5292000	5291999.99519253	5291999.99693719	5291999.62115540	-546346.37500
	1/ 7	-7484400	-7484399.99269844	-7484399.99548502	-7484399.74670860	1205324.00000
	1/ 8	5239080	5239079.99485741	5239079.99673836	5239079.62279320	-1092864.00000
	1/ 9	-1441440	-1441439.99854610	-1441439.99908019	-1441439.95132200	358058.00000
COL 4						
	1/ 1	88200	88199.99992839	88199.99995350	88199.99681171	3889.80078
	1/ 5	-1693440	-1693439.99854476	-1693439.99905093	-1693439.94125680	53050.56250
	1/ 6	10584000	10583999.99053699	10583999.99382806	10583999.64445300	-941790.93750
	1/ 7	-30240000	-30239999.97216026	-30239999.98187263	-30239999.01035000	4063986.00000
	1/ 8	43659000	43658999.95869223	43658999.97329764	43658998.60199600	-7467870.00000
	1/ 9	-31046400	-31046399.97024180	-31046399.98072166	-31046399.02408600	6248209.00000
	1/10	8648640	8648639.99158920	8648639.99456610	8648639.73243110	-1959474.00000
COL 5						
	1/ 1	-194040	-194039.99962565	-194039.99988494	-194039.99245047	12012.50469
	1/ 6	3880800	3880799.99645937	3880799.99765438	3880799.86141850	-398806.93750
	1/ 7	-24948000	-24947999.97698855	-24947999.98475976	-24947999.16374700	3346386.00000
	1/ 8	72765000	72764999.93232919	72764999.95526951	72764997.67813700	-11559803.00000
	1/ 9	-106722000	-106721999.90011085	-106721999.93414663	-106721996.72702000	19084880.00000
	1/10	76839840	76839839.92770875	76839839.95247756	76839837.71936200	-15004450.00000
	1/11	-21621600	-21621599.97957226	-21621599.98661024	-21621599.37570000	4519766.00000
COL 6						
	1/ 1	194040	194039.99981229	194039.99987460	194039.99205053	-26474.01953
	1/ 7	-3991680	-3991679.99619030	-3991679.99744614	-3991679.85452180	639440.43750
	1/ 8	26195400	26195399.97525057	26195399.98341834	26195399.12432600	-4466986.00000
	1/ 9	-77616000	-77615999.92724257	-77615999.95135876	-77615997.57376700	13862053.00000
	1/10	115259760	115259759.89263090?!	115259759.92842036?!	115259756.58597000?!	-21328608.00000
	1/11	-83825280	-83825279.92231215	-83825279.94836591	-83825277.62472500	15952840.00000
	1/12	23783760	23783759.97805123	23783759.98545563	23783759.35066300	-4650262.00000
COL 7						
	1/ 1	-72072	-72071.99992624	-72071.99995025	-72071.99693240	15580.64844
	1/ 8	1513512	1513511.99850374	1513511.99898753	1513511.94400850	-313855.56250
	1/ 9	-10090080	-10090079.99028320	-10090079.99342995	-10090079.66368600	2022861.00000
	1/10	30270240	30270239.97144281	30270239.98073558	30270239.06982900	-5898042.00000
	1/11	-45405360	-45405359.95786702	-45405359.97166081	-45405358.69311400	8633253.00000
	1/12	33297264	33297263.96951975	33297263.97956266	33297263.09193900	-6198050.00000
	1/13	-9513504	-9513503.99136984	-9513503.99424483	-9513503.75204500	1738252.00000

Code Conversion at the Subroutine Level

A commonly encountered matrix problem is shown in skeleton program form below. This code forms part of the solution to a well-known recursive filtering problem. In algebraic terms the required output is a matrix K_i where,

$$K_i = P_i\ H^t\ (H\ P_i\ H^t + R)^{-1}$$

and, at each step,

$$P_i = T\ P_{i-1}\ T^t + Q$$

The original program version on the left assumes each of these matrix functions is computed as a subroutine call. With the MAP-6400 versions on the right, the equivalent operations are carried out using MAP library matrix routines, either as direct replacements for the original calls, or in function list structures to achieve lower MAP/host interaction.

Note that MAP data memories act as a COMMON area for both subroutines, with initialization of the R, Q, and T matrices required only once at the beginning of the run.

Original Host Program

```
      SUBROUTINE PNEXT(P)
      COMMON /FILT/ P,Q,H,..
      ...
      CALL MMUL(P,T,P)
      CALL MMTR(P,P,T)
      CALL MADD(P,P,Q)
      RETURN
      END

      SUBROUTINE KNEXT(K)
      COMMON /FILT/ P,Q,H,..
      ...
      CALL MMTR(TMP,P,H)
      CALL MMUL(K,H,TMP)
      CALL MADD(K,K,R)
      CALL MINV(K,K)
      CALL MMUL(K,TMP,K)
      RETURN
      END

      PROGRAM MAIN
      COMMON /FILT/ P,Q,H,..
      ...
      DO 10 I=1,ITER
      CALL PNEXT(P)
      CALL KNEXT(K)
      CALL XNEXT(X,Z)
      ...
10    CONTINUE
```

MAP User Program

Direct Call Substitution

```
      SUBROUTINE PNEXT(P)
      COMMON /MAP/ PP,QQ,HH,...
      ...
      CALL MRMUL(PP,TT,PP)
      CALL MRMTR(PP,PP,TT)
      CALL MRADD(PP,PP,QQ)
      RETURN
      END
```

- Define subroutine PNEXT using MAP calls.
- Real matrix multiply
- Matrix x matrix transpose
- Real matrix add
- Similar modifications to KNEXT
- Define main program

```
      PROGRAM MAIN
      COMMON /MAP/ PP,QQ,HH,...
      ...
      CALL MPINIT(PP,QQ,HH,...
      DO 10 I=1,ITER
      CALL PNEXT(P)
      CALL KNEXT(K)
      ...
10    CONTINUE
```

- Open MAP, initialize data areas
- Call host subroutines in usual way

Function List Substitution

```
      PROGRAM MAIN
      COMMON /MAP/ PP,QQ,HH,...
      ...
      CALL MPINIT(PP,QQ,HH,...
      CALL MPBFL(PNXT)
        CALL MRMUL(PP,TT,PP)
        CALL MRMTR(PP,PP,TT)
        CALL MRADD(PP,PP,QQ)
      CALL MPEFL(PNXT)
      ...
      DO 10 I=1,ITER
      CALL MPXFL(PNXT)
      CALL MPXFL(KNXT)
      ...
10    CONTINUE
```

- Define main program including MAP function lists
- Open MAP, initialize data areas
- Begin function list definition
- Real matrix multiply
- Matrix x matrix transpose
- Real matrix add
- End function list PNXT
- Call host defined function lists
- Function list execution in MAP unburdens host CPU

executed. These routines are simply caretakers for the true MAP resident array processing routines and their role is to pass sets of parameters to the MAP via a host resident driver program.

When an array function command is received by the MAP Executive, execution of the MAP object module is initiated in the appropriate arithmetic or I/O processor.

Extended Executive Features

The MAP Executive is smart enough to handle more than just single calls from the host. Totally independent parallel processing can be achieved with the definition of function lists—strings of array function calls that are constructed as a group in the host program. Function lists may be executed repetitively using the WHILE and FOR commands, conditionally using the IF and CASE OF commands, or in a single pass with an unconditional command. Control calls can also be imbedded in function lists, thereby providing nested function list capabilities.

Innovation in Array Processing

CSPI was formed in 1968 to meet the growing requirement for high-speed array processing systems in signal processing, data reduction and other computationally intensive fields.

The MAP Array Processor product line, introduced in 1974, represented a bold departure from traditional processor architecture.

Today, several hundred MAPs are in service worldwide—in labs, aboard ship, in airborne systems, and installed in diverse OEM equipment. Wherever the demand for advanced array processing exists, CSPI stands ready to fill the need.

cspi CSP Inc.

209 Middlesex Turnpike, Burlington, MA 01803 Tel: 617-272-6020 TWX: 710-332-0835

The leader in Digital Signal Processing